AGVs Picking Path Planning Considering Mixed Storage Strategy in Intelligent Warehouse

Yanju Zhang [a*], Qinggang Yang [a]

[a.] School of business administration, Liaoning Technical University, Huludao, Liaoning, 125105, China
zhangyanju@lntu.edu.cn*, 576870146@qq.com

Abstract. To improve the picking efficiency of orders in intelligent warehouses, this article conducts research on the AGV picking path planning problem. Firstly, a mixed storage strategy is introduced based on order characteristics, and a mathematical model is constructed with the objective of minimizing the total time for AGVs to complete all orders. Then, an improved Q-Learning algorithm with a greedy parameter and embedded conflict resolution strategy is proposed to obtain the optimal conflict-free picking path solution. Finally, through numerical comparison and analysis of examples, it is found that compared with existing path planning algorithms, the proposed algorithm reduces the total time for AGVs to complete all orders by 13.79% and 27.82%, respectively. The comparison of indicators such as the number of AGVs used and the proportion of waiting time due to path conflicts verifies that the proposed algorithm and mixed storage strategy can effectively alleviate congestion, reduce the length of driving paths, and improve picking efficiency.

Keywords: intelligent warehouse, mixed storage strategy, conflict-free path planning, improved Q-Learning algorithm

1 INTRODUCTION

With the growth of e-commerce and internet mobile payments, online shopping has emerged as a primary consumption mode, causing e-commerce enterprises to handle vast order volumes daily. Consumer demand is also becoming more personalized and diverse, resulting in orders featuring multiple varieties, small batches, and numerous batches[1].

The "cargo-to-person" picking system supported by intelligent technology has gradually become the focus of scholars' research. Masae et al proposed a Eulerian graph and a dynamic programming process based on optimal picking paths in the warehouse layout with leaf shapes[2]. Zhuang et al introduced an adaptive large neighborhood search in automated warehousing to address mixed-integer programming. Compared to a company's pre- and post-implementation practices, this method reduced shelf movement by up to 62%[3]. Lin et al used a mixed PSO-SA algorithm for AGV path planning in intelligent warehouses, overlooking potential AGV collisions. This oversight may reduce the algorithm's practical effectiveness and raise safety

V. Vasilev et al. (eds.), *Proceedings of the 2024 5th International Conference on Management Science and Engineering Management (ICMSEM 2024)*, Advances in Economics, Business and Management Research 306,
https://doi.org/10.2991/978-94-6463-570-6_119

issues[4] .Guo et al planned conflict-free paths based on a fusion of an improved A* algorithm and the dynamic window approach[5]. However, the A* algorithm has poor adaptability and is prone to deadlocks in dynamic and complex environments. Zhuo et al employed Q-learning for real-time path planning for automatic guided vehicles in container terminals[6]. However, the traditional Q-Learning algorithm suffers from slow convergence during the learning process.

Few scholars have examined the impact of storage strategies on AGV paths in intelligent warehouses. This paper considers mixed storage strategies' effects on AGV picking paths and introduces a Q-Learning algorithm with greedy parameters and conflict resolution strategies to mitigate path conflicts among multiple AGVs.

2 PROBLEM DESCRIPTION AND MODEL BUILDING

2.1 Problem Description

Figure 1 illustrates the order picking process in the intelligent warehouse system. Orders are generated from the online retail platform, and shelves are assigned to AGVs for transportation. AGVs move to the target shelves, transport them to picking stations, and then return them to their original locations. Figure 2 shows a simplified warehouse layout using the grid method, with shelves represented by black grids and picking aisles by white grids. AGV handling tasks are divided into three stages: M1 (moving to target shelf), M2 (transporting and picking), and M3 (returning to storage).

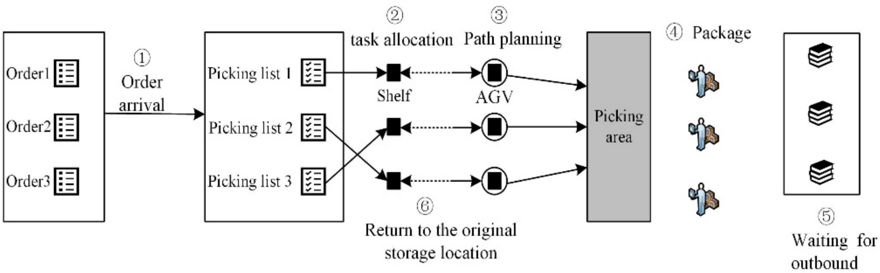


Fig. 1. Order picking process of intelligent warehouse system（Drawn by the author）

To further understand the order picking process in the intelligent warehouse system, this study provides formal definitions of relevant elements.

Definition 1 (Warehouse). Discretely meshed M warehouses that $M[i,j]$ represent i a grid of rows and j columns. The value of the grid can be 0 or 1, indicating whether there are shelves at this location.

Definition 2 (AGV). An AGV is defined as $q=\{M_q[i,j],w,C_q\}$, the $M_q[i,j]$ current position of the AGV in the warehouse and w the maximum load-bearing weight of the AGV. The status of the AGV $C_q\in\{0,1\}$, if it $C_q=1$ means that the AGV is performing the handling work of the shelf and is in a busy state, $C_q = 0$ it means that the AGV currently has no handling tasks and is in a free state.

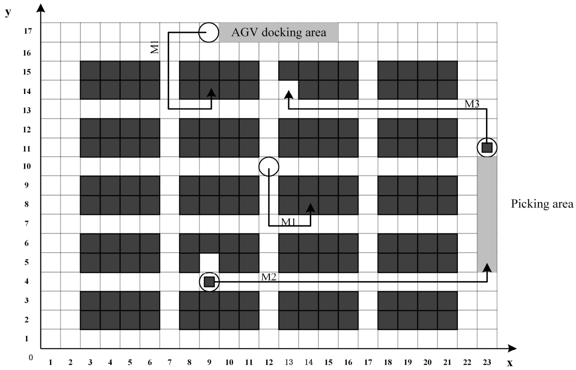


Fig. 2. Layout diagram of intelligent warehouse system based on grid method（Drawn by the author）

Definition 3 (Shelf). A shelf is defined as $k = \{M_k[i,j], C_k\}$, indicating the $M_k[i,j]$ location of the shelf. The status of the shelf $C_k \in \{0,1\}$, if $C_k = 1$ it means that the shelf is being carried by the AGV and is in a busy state, it $C_k = 0$ means that the shelf is not being carried by the AGV and is in a free state.

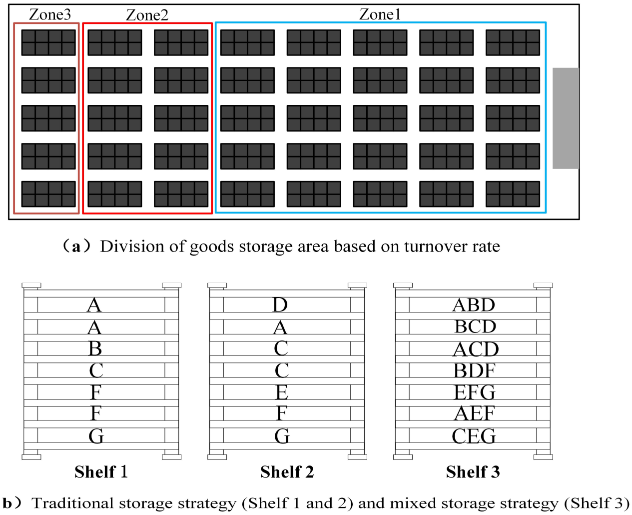
Definition 4 (Picking Station). A picking station is defined as $v = \{M_v[i,j], I_v, C_v\}$ the $M_v[i,j]$ location of the picking station, the number of the picking station is I_v indicated, and the status of the picking station $C_v = \{0,1\}$, if $C_v = 1$ means that the picking station is performing a goods picking task and is busy, if $C_v = 0$ means that the picking station has no picking task and is free.

Definition 5 (Path). A route is defined as $R = \{P_{start}, P_{end}, t_{start}, R^*\}$, where P_{start} and P_{end} represent the start and end point of the route, respectively. t_{start} represents the departure time of the path. $R^* = \{p_1, p_2, \cdots, p_n\}$,it is a sub-segment sequence corresponding to the path, which consists of a series of sub-segments $p = \{t^q_{start}, t^q_{end}, M_q[i,j]_{start}, M_q[i,j]_{end}, M_q[i,j]_{dist}\}$, and each item in the sub-segment represents the start time and end time of the sub-segment, the coordinates of the start point of the sub-segment, the coordinates of the end point of the sub-segment, and the coordinate difference between the start point and the end point.

The following assumptions are made: (1) Order and product information are known before sorting; (2) The fulfillment warehouse can meet the demand of all orders, and there is no shortage of goods; (3) The weight of the movable shelves is within the carrying capacity of the AGVs; (4) Each AGV can only transport one shelf at a time; (5) All AGVs have the same specifications and travel at a unit speed of 1 cell per second.

2.2 Mixed Storage Strategy

Fulfillment warehouses aim to expedite order processing and minimize non-productive walking, favoring mixed storage. Figure 3(a) sorts goods by turnover rate, with Zone 1 having high-turnover goods near picking stations for high utilization, Zone 2 having lower-turnover goods further away with average utilization, and Zone 3 for the rest. Figure 3(b) shows Shelf 3 using mixed storage, grouping related goods, while Shelves 1 and 2 use traditional strategies with one product per layer, leading to weaker associations and longer picking time.



(a) Division of goods storage area based on turnover rate

(b) Traditional storage strategy (Shelf 1 and 2) and mixed storage strategy (Shelf 3)

Fig. 3. Division of goods area and comparison of storage strategies based on turnover rate (Drawn by the author)

2.3 Model Building

Based on the problem, definitions, and assumptions, here are the sets, parameters, and decision variables in the model:

$Q=\{1,2,\cdots,q\}$ is the set of AGVs. $O=\{1,2,\cdots,o\}$ is the set of orders.

$K=\{1,2,\cdots,k\}$ is the set of shelves. $V=\{1,2,\cdots,v\}$ is the set of picking stations. l is the list of orders to be picked, t_k^q is the time for an AGV to travel from its unloaded position to the target shelf; t_{kv}^q is the time for picking at the target shelf; t_{vk}^q is the time for the AGV to return from the picking station to the shelf. x_{kv}^q is a 0-1 variable indicating whether the AGV transports a shelf to the picking station (1) or not (0);

y_k^q indicates if the AGV is scheduled to the shelf (1) or not (0); z_{vk}^q indicates if the AGV returns the shelf to its original location (1) or not (0).

1) Objective function:

$$\min \sum_{q \in Q} \left(t_k^q + t_{kv}^q + t_{vk}^q \right) \tag{1}$$

2) Constraints:

$$t_k^q - \sum_{q \in Q} y_k^q = 0, \ \forall k \in K \tag{2}$$

$$t_{kv}^q - \sum_{v \in V} x_{kv}^q = 0, \ \forall k \in K, v \in V \tag{3}$$

$$\sum_{v \in V} x_{kv}^q = 1, \ \forall q \in Q, k \in K \tag{4}$$

$$\sum_{k \in K} x_{kv}^q - \sum_{k \in K} z_{vk}^q = 0, \ \forall q \in Q, v \in V, k \in K \tag{5}$$

$$x_{kv}^q - \sum_{k \in K} z_{vk}^q = 0, \ \forall q \in Q, v \in V \tag{6}$$

$$x_{kv}^q, y_k^q, z_{vk}^q \in \{0,1\}, \ \forall q \in Q, k \in K, v \in V \tag{7}$$

The objective function aims to minimize the total completion time of all AGVs. Constraint (2) ensures AGV handling time is 0 if it hasn't moved to the shelf. Constraint (3) applies if the AGV hasn't transported the shelf to the picking station. Constraint (4) limits the AGV to transporting the shelf to one picking station. Constraint (5) ensures AGVs return shelves to their original locations after picking. Constraint (6) excludes the need for returning shelves if not transported to the picking station. Constraint (7) defines the decision variable constraints.

3 PATH PLANNING ALGORITHM BASED ON CONFLICT RESOLUTION STRATEGY

3.1 Conflict Resolution Strategy

During the driving process, AGVs mainly encounter three types of conflicts: intersection conflicts, opposing conflicts, and rear-end conflicts, as shown in Figure 4.

(1) Opposing Conflict Strategy: If two AGVs travel opposite directions on the same path, causing a collision, the AGV with the shorter path gets priority. If paths are the same length, priority is randomly assigned.

(2) Intersection Conflict Strategy: When AGVs arrive at the same intersection simultaneously, the one with the shorter path gets priority. If paths are equal, priority is randomly chosen.

(3) Rear-end Conflict Strategy: To prevent AGV1 from rear-ending stationary AGV2, the later-arriving AGV waits until the other leaves the grid.

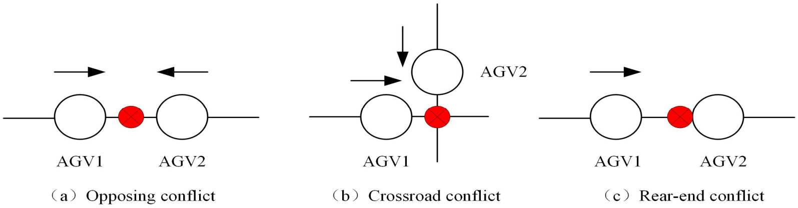


Fig. 4. Types of AGV Conflicts（Drawn by the author）

3.2 Improved Q-Learning Path Planning Algorithm

Compared to A* and D*[7]algorithms, Q-learning[8] offers simplicity, adaptability, and self-learning in unknown environments. To avoid local optima, we introduce an improved Q-Learning with a conflict resolution strategy and a greedy parameter $\alpha\left(0 \le \alpha \le 100\%\right)$. With probability α, the AGV follows the assigned path; with $1-\alpha$, it chooses randomly. The simplified algorithm steps are as follows:

Step 1: Initialize reward matrix Re and Q^*-value table;

Step 2: Check if the maximum learning iterations have been reached. If not, continue; if so, proceed to Step 8;

Step 3: Initialize relevant parameters;

Step 4: Check if the final goal has been reached. If not, continue; if so, proceed to Step 8;

Step 5: Select the action state;

Step 6: Determine the instantaneous reward based on equations (8), (9), and (10);

Step 7: Update the Q^*-value and decay relevant parameters;

Step 8: Check if the Q^*-value table has converged. If not, return to Step 3 and continue; if so, proceed;

Step 9: Adjust the path based on the conflict resolution strategy;

Step 10: End of the algorithm. Output the optimal conflict-free path R.

In Q-Learning, reward Re_1 is given upon reaching the target. To hasten arrival, we add reward Re_2 for actions closer to the target. The formula is:

$$Re_2 = \begin{cases} 10, & \Delta dist < 0 \\ -10, & \Delta dist > 0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

In the formula, $\Delta dist$ represents the difference in Manhattan distance between the current grid position $M_q\left[i,j\right]$ of the AGV, the previous grid position $M_q\left[i,j\right]'$, and the final grid position $M_{end}[i,j]$. The final position $M_{end}[i,j]$ can be either the picking station $M_v[i,j]$ or $M_k[i,j]$. The formula is as follows:

$$\Delta dist = \left(\left|M_q[i]-M_{end}[i]\right|+\left|M_q[j]-M_{end}[j]\right|\right)-\left(\left|M_q[i]^{'}-M_{end}[i]\right|+\left|M_q[j]^{'}-M_{end}[j]\right|\right) \tag{9}$$

Besides instant reward Re_1 for reaching the target and reward Re_2 for approaching it, the AGV gets a -1 penalty for each step to hasten its arrival. The overall reward-penalty formula is:

$$Re = Re_1 + Re_2 - 1 \tag{10}$$

4 EXPERIMENT AND ANALYSIS

4.1 Data Description

A 40m×40m double-sided shelf layout with 768 shelves, each having 6 layers for 60 products, is set up in a smart warehouse. To validate our proposed path planning method, we use the parameters in Table 1. The experiments are run on a laptop with Inter core AMD (3.2Ghz), 16GB RAM, and Windows 11, using Matlab 2022a for algorithm solving.

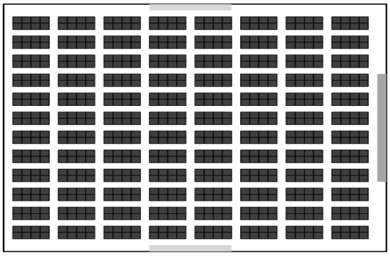


Fig. 5. The diagram of warehouse layout（Drawn by the author）

Table 1. Parameters of the example（Produced by the author）

Example	Order quantity	The quantity of the product in demand
1	100	224
2	200	476
3	500	927
4	800	1753
5	1000	1889

4.2 Mixed Storage and Analysis of Algorithm Effectiveness

A comparative analysis of the improved Q-Learning algorithm (mixed storage) for AGVs is conducted, focusing on total distance (TD), total time (TT), and task efficiency (TE). The baseline for comparison is the standard Q-Learning and an improved A* algorithm. The results, as shown in Table 2, indicate that with increasing order quantities, the improved Q-Learning algorithm outperforms the others, reducing transportation task completion time by an average of 13.79% and 27.82% compared to the other algorithms.

Table 2. The comparison data between the proposed algorithm and the existing algorithms

Example	Improved Q-Learning (mixed storage)			Q-Learning			Improved A* algorithm		
	TD	TT	TE	TD	TT	TE	TD	TT	TE
1	88	92	95.65%	106	112	94.64%	142	162	87.65%
2	169	189	89.42%	186	204	91.18%	272	295	92.20%
3	397	431	92.11%	404	417	96.88%	494	582	84.88%
4	693	729	95.06%	741	783	94.64%	853	944	90.36%
5	958	1003	95.51%	1165	1319	88.32%	1233	1403	87.88%
Average	461	488.8	94.31%	520.4	567	91.78%	565.2	677.2	83.46%

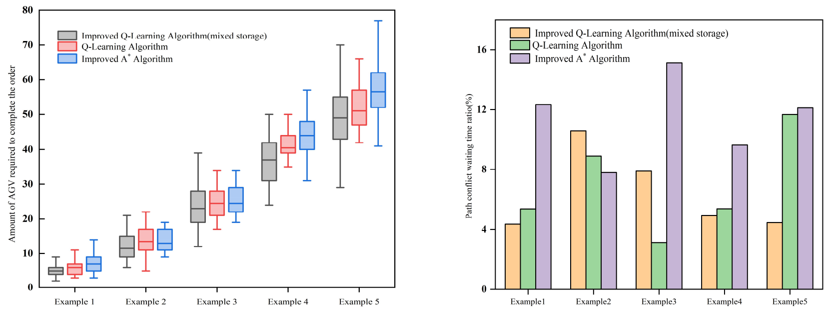


Fig. 6. The comparison of the required number of AGVs for order completion and the proportion of waiting time due to path conflicts (Drawn by the author)

As shown in Figure 6 (left), the improved Q-Learning algorithm performs the best in the five test cases, requiring the smallest number of AGVs to complete all orders. This is because the improved Q-Learning algorithm takes into account a mixed storage strategy, which, with its support, can reduce the number of AGVs used. The Figure 6 (right) demonstrates the proportion of waiting time due to path conflicts for different algorithms in different test cases. As the number of orders increases, the proposed algorithm gradually demonstrates its advantages. Especially when the order volume increases to 800 orders, it can effectively mitigate the path conflicts among AGVs and save time for order completion, which also illustrates the necessity of the mixed storage strategy.

5 CONCLUSIONS

This article studies AGV path planning in intelligent warehouses, considering factors like multi-shelf storage and path conflicts. A mathematical model is established to minimize total order completion time. An improved Q-Learning algorithm with a path conflict resolution strategy is proposed. Experimental results show that the algorithm reduces AGV path conflicts and enhances picking efficiency. This study not only advances AGV path planning theories but also provides practical guidance for enterprises to improve picking efficiency. Furthermore, considering various practical factors makes the study more meaningful.

REFERENCES

1. WINKELHAUS S, GROSSE E H. (2020) Logistics 4.0: a systematic review towards a new logistics system [J]International Journal of Production Research,58(1):18-43. https://doi.org/10.1080/00207543.2019.1612964.
2. Masae M, Glock C H, Vichitkunakorn P. (2021) A method for efficiently routing order pickers in the leaf warehouse[J]. International Journal of Production Economics, 234: 108069. https://doi.org/10.1016/j.ijpe.2021.108069.
3. ZHUANG Y, ZHOU Y, YUAN Y, et al. (2022) Order picking optimization with rack-moving mobile robots and multiple workstations [J]. European Journal of Operational Research,300(2): 527-44. https://doi.org/10.1016/j.ejor.2021.08.003.
4. Lin S, Liu A, Wang J, et al. (2023) An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse[J]. Journal of Computational Science,67:101938. https://doi.org/10.1016/j.jocs.2022.101938.
5. Guo T, Sun Y, Liu Y, et al. (2023) An Automated Guided Vehicle Path Planning Algorithm Based on Improved A* and Dynamic Window Approach Fusion[J]. Applied Sciences,13(18): 10326. https://doi.org/10.3390/app131810326.
6. ZHOU, Pengfei; LIN, Li; KIM, Kap Hwan. (2023) Anisotropic Q-learning and waiting estimation based real-time routing for automated guided vehicles at container terminals. Journal of Heuristics, 1-22.
7. XIE, Kaili; QIANG, Jie; YANG, Haitao. (2020) Research and optimization of d-start lite algorithm in track planning. IEEE Access, 8: 161920-161928. https://doi.org/10.1109/ACCESS.2020.3021073.
8. TAN, Tao; XIE, Hong; FENG, Liang. (2024) Q-learning with heterogeneous update strategy. Information Sciences, 656: 119902. https://doi.org/10.1016/j.ins.2023.119902.